

Práctica 3 (2 ptos)

Planificación

1.– Objetivo

En esta práctica se va a profundizar en el multiproceso de UNIX realizando un planificador (scheduler) de procesos a alto nivel, basado en señales, que implementa una disciplina turno rotativo (round robin).

2.– Planificador a nivel de usuario `usersched`

`USERSCHED(1)`

`USERSCHED(1)`

NAME

`usersched` – Scheduler de procesos a nivel de usuario

SYNOPSIS

`usersched [configfile]`

DESCRIPTION

El programa `usersched` lanza los procesos que le han sido configurados y realiza un reparto del tiempo de CPU entre los procesos con una disciplina round-robin.

El fichero de configuración `configfile` es opcional, de no estar presente se leerá la información de configuración de la entrada estándar, siguiendo el mismo formato que el fichero.

Formato del fichero de configuración: Consta de una línea por cada programa a ejecutar. Cada línea es de la siguiente forma:

```
cuanto nombreprograma opcion1 opcion2 opcion3...
```

Donde:

`cuanto`: La cantidad de tiempo consecutiva que se le asignará a este proceso antes de pasar a otro (entero en segundos).

`nombreprograma opcion1 opcion2 opcion3...`: El programa a ejecutar junto con sus opciones, que pueden ser un número indeterminado y diferente para cada programa.

El programa retira del ciclo de round-robin los procesos que vayan finalizando su ejecución. `usersched` termina cuando todos los procesos a planificar han acabado.

3.– Ficheros

En el directorio `$(HOME)/solucion/prac3` debe encontrarse un `Makefile` así como todos los ficheros `.c` y `.h` necesarios para crear `usersched`. La acción por defecto del `Makefile` (la cual debe funcionar con solo hacer `make` en ese directorio) debe ser crear `usersched`.

Se sugiere emplear la librería `fragmenta.a` creada en la práctica 2.

En el directorio `$(HOME)/../ejemplos/prac3` se puede encontrar un ejemplo de `usersched`.

Para la corrección de la práctica se borrarán todos los ejecutables y se recompilará mediante el `Makefile`.

4.– Aplicaciones, funciones y llamadas al sistema útiles

`kill(1)`, `kill(2)`, `raise(3)`, `signal(2)`, `signal(7)`,
`sigaction(2)`, `sleep(3)`, `nanosleep(2)`, `pause(2)`, `alarm(2)`.